

# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Il contesto applicativo</b>	<b>4</b>
<b>2 Descrizione del lavoro</b>	<b>5</b>
2.1 Livello black box . . . . .	6
2.2 Livello sottosistemi . . . . .	7
2.3 Livello moduli . . . . .	10
<b>3 Tecnologie utilizzate</b>	<b>14</b>
3.1 Sensori MEMS . . . . .	14
3.2 Ultra-wideband . . . . .	14
<b>4 Elaborazione dei dati acquisiti e algoritmo di data fusion</b>	<b>15</b>
4.1 Rumore dei dati grezzi . . . . .	15
4.2 Tecniche di elaborazione . . . . .	15
4.3 Algoritmo di data fusion . . . . .	15
<b>5 L'implementazione</b>	<b>16</b>
<b>6 Analisi e validazione dei risultati</b>	<b>17</b>
<b>7 Conclusioni e prospettive future</b>	<b>18</b>

# Elenco delle figure

1	Planimetria originale . . . . .	1
2	Planimetria alterata . . . . .	1
3	Rete allo step 1 dell'esploratore . . . . .	2
4	Rete allo step 1 dell'esploratore . . . . .	2
5	Rete allo step 3 dell'esploratore . . . . .	2
6	Rete allo step 4 dell'esploratore . . . . .	2
7	Rete allo step 5 dell'esploratore . . . . .	3
8	Rete allo step 6 dell'esploratore . . . . .	3
2.1	Rappresentazione dei livelli d'astrazione utilizzati per descrivere il sistema . . . . .	5
2.2	Rappresentazione del sistema come black box . . . . .	6
2.3	Rappresentazione del sistema in sottosistemi . . . . .	7
2.4	Ambiguità nella georeferenziazione di un secondo nodo utilizzando soltanto la distanza dal precedente . . . . .	8
2.5	Rappresentazione dei sottosistemi in moduli . . . . .	10
2.6	Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 1 . . . . .	11
2.7	Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 2 . . . . .	12
2.8	Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 3 . . . . .	12
2.9	Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 4 . . . . .	13

# Introduzione

Il lavoro di questa tesi si colloca nel progetto aziendale Tekne per la realizzazione di un sistema di geolocalizzazione di operatori in contesti privi di segnale GPS. Il sistema permette ad un “esploratore” di creare dinamicamente una rete di nodi all’interno di zone nelle quali il segnale GPS è assente o comunque debole. Questa rete verrà poi ampliata ed utilizzata dagli operatori successivi ad esso per geolocalizzarsi all’interno della zona ed intervenire in maniera ottimale.

Uno scenario esemplificativo è quello di un vigile del fuoco che interviene in un edificio per soccorrere una persona. Tale operatore è considerato un esploratore, poiché è il primo ad intervenire e la planimetria dell’edificio risulta essere ignota e/o cambiata a seguito dell’evento disastroso (Fig.1 e Fig.2).

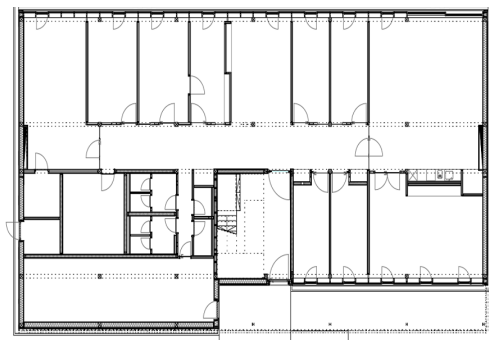


Figura 1: Planimetria originale

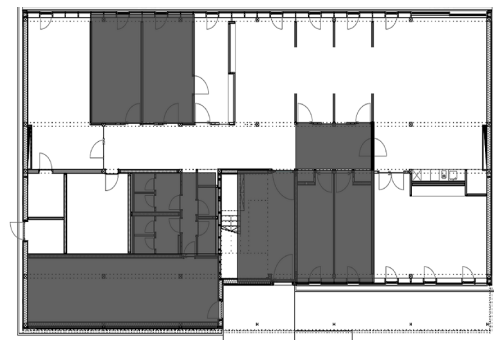


Figura 2: Planimetria alterata

I rettangoli in grigio (Fig.2) rappresentano aree non più accessibili dell'edificio, mentre le mura interrotte nuovi percorsi creati a causa dei crolli. Durante tutta la fase di scouting, l'esploratore posizionerà un'ancora (nodo) ogni 20 metri approssimativamente e ogni qualvolta la precedente risulti non essere più in line-of-sight (linea visiva). Così facendo si "lascerà dietro una scia di briciole" che gli permetteranno di orientarsi all'interno dell'edificio, di eseguire il percorso all'inverso o di ricevere supporto da un'ulteriore operatore. Nelle figure seguenti, si può notare come la rete cresca man mano che l'esploratore avanza all'interno dell'edificio.

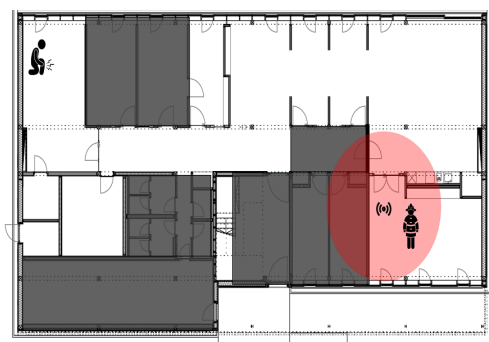


Figura 3: Rete allo step 1 dell'esploratore

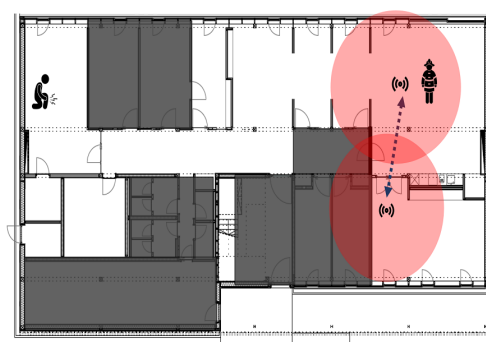


Figura 4: Rete allo step 1 dell'esploratore

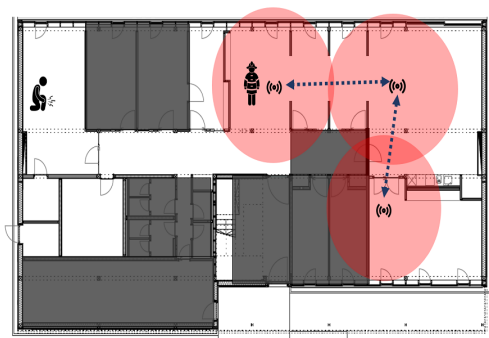


Figura 5: Rete allo step 3 dell'esploratore

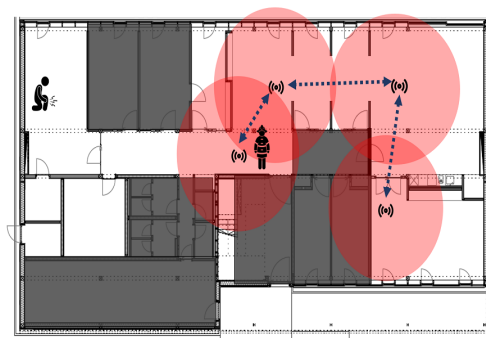


Figura 6: Rete allo step 4 dell'esploratore

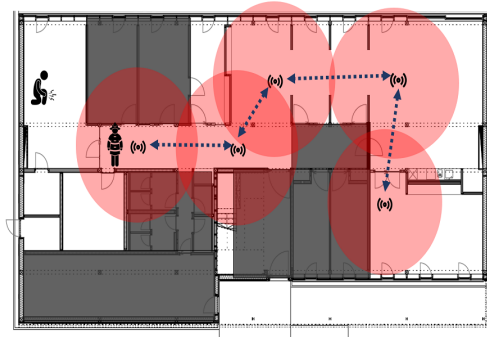


Figura 7: Rete allo step 5 dell'esploratore

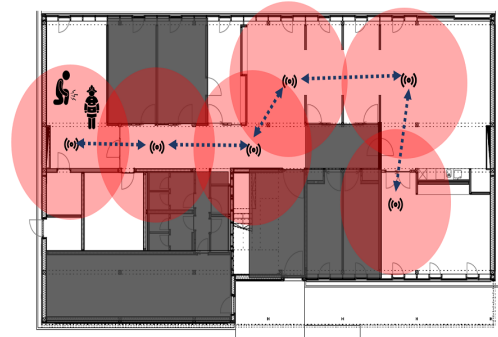


Figura 8: Rete allo step 6 dell'esploratore

Una volta creata l'infrastruttura, gli operatori potranno comunicare e condividere informazioni come posizione e stato.

La presente tesi è così strutturata:

**Capitolo 1:** Viene descritto il problema della geolocalizzazione indoor e lo stato dell'arte.

**Capitolo 2:** Viene descritto il lavoro svolto con un approccio top down.

**Capitolo 3:** Vengono illustrate le tecnologie utilizzate (sensori MEMS e UWB) e le problematiche legate ad esse.

**Capitolo 4:** Viene descritto l'elaborazione dei dati grezzi e l'algoritmo di data fusion utilizzato.

**Capitolo 5:** Si riporta l'implementazione software e hardware dei sottosistemi.

**Capitolo 6:** Si analizzano i risultati ottenuti.

# **Capitolo 1**

## **Il contesto applicativo**

–Aggiungere motivazioni del lavoro, stato dell’arte della geolocalizzazione indoor e quali problemi si vogliono risolvere–

## Capitolo 2

### Descrizione del lavoro

Per comprendere la soluzione proposta e il sistema ideato utilizzeremo un approccio top-down. Partendo da un livello d'astrazione tale in cui emergono solo i requisiti funzionali e il sistema è rappresentato da una *black box*, scenderemo ad un livello che ci permetterà di affinare i requisiti iniziali e mostrare i *sottosistemi* che lo costituiscono. Infine nell'ultimo livello avremo una visione dei *moduli* che compongono i sottosistemi e che sono stati realizzati nel contesto di questa tesi.

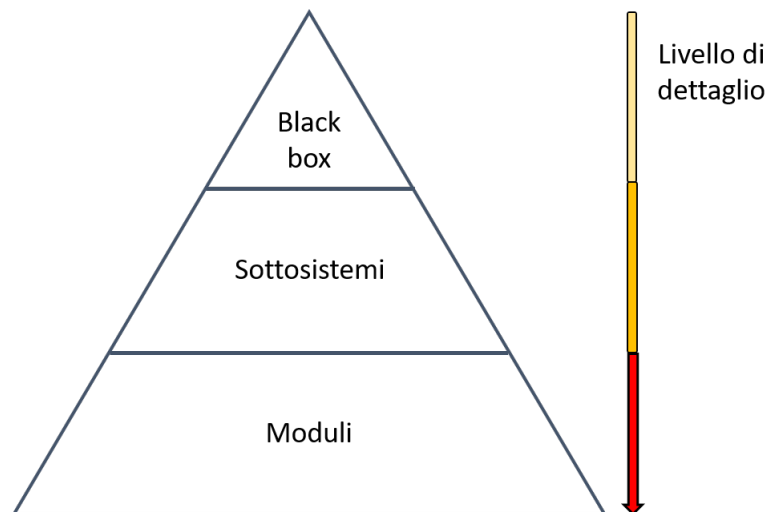


Figura 2.1: Rappresentazione dei livelli d'astrazione utilizzati per descrivere il sistema

## 2.1 Livello black box

Considerando il sistema in questione come una black box (Fig.2.2) e l'infrastruttura di rete in modo astratto, i due macro-requisiti funzionali sono rispettivamente:

- **R1:** Geolocalizzare l'operatore
- **R2:** Trasmettere messaggi predefiniti come stato della vittima e codici d'emergenza.

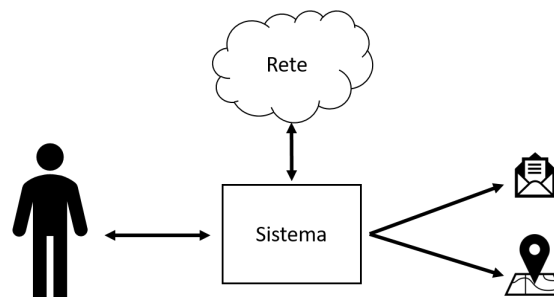


Figura 2.2: Rappresentazione del sistema come black box

Nelle fasi primordiali del progetto, si è scelto di allocare la maggior parte delle risorse lavorative nel completamento di **R1** lasciando ad una fase successiva lo sviluppo di **R2**, per questo motivo d'ora in avanti considereremo come unico requisito la geolocalizzazione dell'operatore. *R1* può essere suddiviso in due requisiti più specifici:

- **R1.1:** Determinare la posizione di un operatore all'interno della rete
- **R1.2:** Identificare il cammino minimo da un nodo ad un altro

Quest'ultimo rappresenta sia la possibilità da parte dell'operatore di eseguire il percorso all'inverso, sia la possibilità che venga raggiunto da una squadra di supporto. A questo punto possiamo scendere al livello d'astrazione successivo (2.1).



## 2.2 Livello sottosistemi

Come già accennato nell'introduzione, la rete verrà costruita dinamicamente da un'esploratore e man mano che egli avanza verrà ampliata aggiungendo nuovi nodi. Realizzare una rete del genere introduce numerose problematiche, alcune strettamente legate alla tecnologia utilizzata (vedi cap.3) altre alle caratteristiche richieste. Nello specifico la problematica sulla quale il lavoro di questi tesi si è concentrato è la progettazione e lo sviluppo dei sottosistemi destinati alla georeferenziazione della rete, ovvero l'attribuzione dell'informazione riguardante la dislocazione geografica dei nodi all'interno della rete.

In questo contesto i sottosistemi individuati nella fase di progettazione sono rappresentati dalla seguente figura:

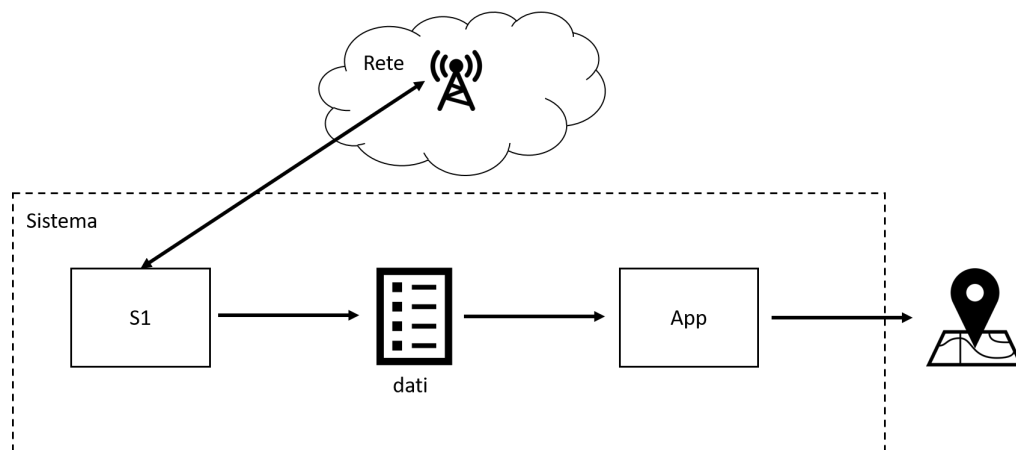


Figura 2.3: Rappresentazione del sistema in sottosistemi

Ognuno dei quali con la propria responsabilità:

- **S1**: ha il compito di ricavare tramite la rete e altri sensori i dati necessari
- **App**: ha il compito di ricevere i dati da S1, elaborarli e infine fornire la posizione del nuovo nodo rispetto alla rete

La soluzione al problema si costruisce per iterazione georeferenziano i singoli nodi nel momento in cui vengono aggiunti dall'operatore. Con riferimento all'esempio illustrato precedentemente (vedi Fig.3- Fig.8) consideriamo lo step 2.

Per ipotesi supponiamo che il primo nodo sia già georeferenziano, nel momento

in cui l'operatore risulti essere al limite della line-of-sight e/o della distanza di sicurezza (20 mt) piazzerà il secondo nodo. Per georeferenziarlo (rispetto al primo) abbiamo bisogno di due informazioni fondamentali:

- La distanza tra i due nodi
- L'angolo tra i due nodi

Per mantenere il livello d'astrazione attuale ci basta sapere che le caratteristiche della tecnologia utilizzata nell'implementazione della rete, fa sì che le singole celle abbiano un raggio d'azione all'interno del quale il sottosistema *SI* può calcolare la distanza tra l'operatore e il centro della cella di appartenenza.

Tale informazione non è però sufficiente, infatti ci sono infiniti punti sulla circonferenza con centro nel primo nodo e raggio pari alla distanza. Per poter georeferenziare in modo univoco il secondo nodo abbiamo bisogno di trovare anche l'angolo in riferimento al primo nodo, tale compito non è banale e tanto meno le metodologie univoche e perfette.

La figura seguente rappresenta un tentativo di georeferenziare il secondo nodo utilizzando soltanto la distanza tra i due nodi.

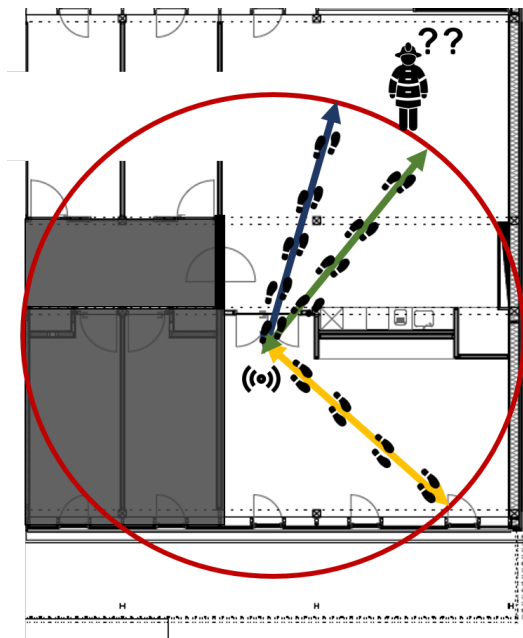


Figura 2.4: Ambiguità nella georeferenziazione di un secondo nodo utilizzando soltanto la distanza dal precedente

Nell'esempio appena proposto, abbiamo mostrato solo tre delle possibili infinite posizioni del secondo nodo. Assumiamo che il punto corretto sia l'intersezione tra il vettore in blu e la circonferenza. In tal caso un'ambiguità con il vettore in verde potrebbe essere accettata in quanto si discosta di pochi metri dalla reale posizione, ben diversa sarebbe un'ambiguità con il vettore in giallo che renderebbe l'informazione del tutto errata e il sistema disinformatore.

Per il momento ci basta sapere che l'algoritmo utilizzato nel contesto di questa tesi (vedi cap.4.3), esegue un campionamento durante il tragitto tra un nodo e l'altro combinando i dati inerenti alla distanza dell'operatore con quelli relativi agli angoli lungo il percorso. Nel prossimo paragrafo ci porteremo al livello d'astrazione più basso della nostra piramide (vedi 2.1) dettagliando i moduli che compongono i due sottosistemi e illustrando come questi intendono risolvere il problema di georeferenziazione appena esposto.

## 2.3 Livello moduli

Prima di "aprire" i due sottosistemi è bene specificare che utilizzeremo il termine *modulo* per riferirci sia a componenti hardware che software. Questo abuso di notazione ci permetterà di mostrare con un unico livello d'astrazione tutti i moduli progettati e realizzati al fine di risolvere il problema della georeferenziazione dei nodi.

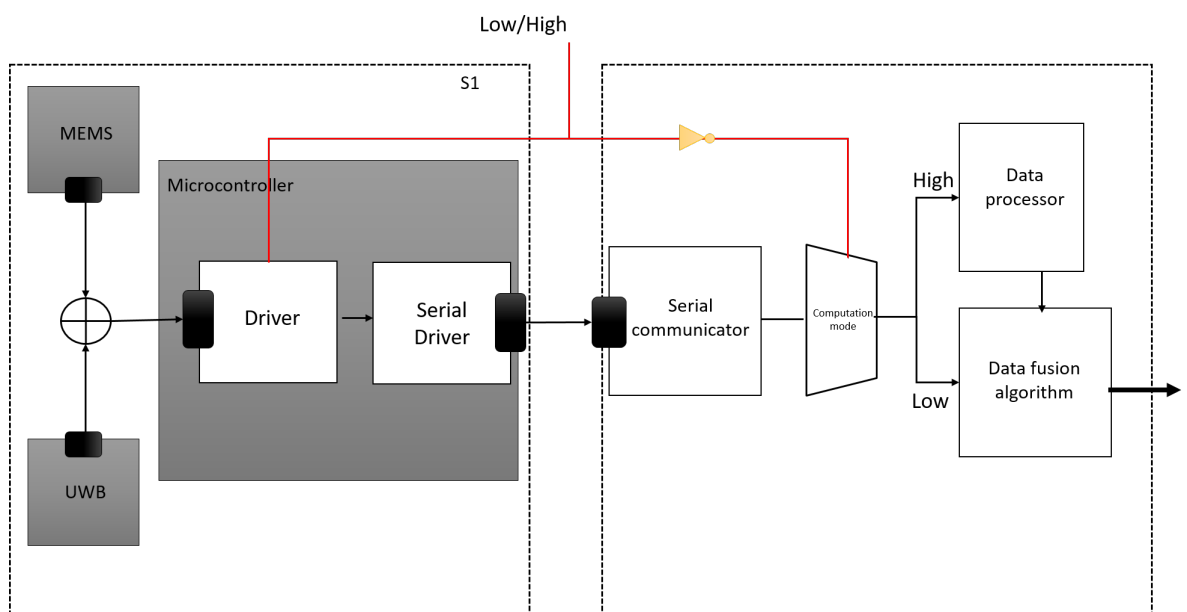


Figura 2.5: Rappresentazione dei sottosistemi in moduli

Per esplicitare al meglio i compiti dei singoli moduli e mantenere l'attuale livello d'astrazione, fermo restando che tutti i dettagli tecnici verranno forniti nei capitoli successivi, mostreremo il flusso di dati generato in uno qualsiasi degli intervalli di campionamento lungo il tragitto dell'operatore tra un nodo e l'altro.

Il flusso inizia nel momento in cui il *driver* acquisisce le informazioni dai moduli *MEMS* e *UWB* riguardanti la distanza, la velocità angolare e l'accelerazione lineare dell'operatore. Come mostrato dalla figura seguente.

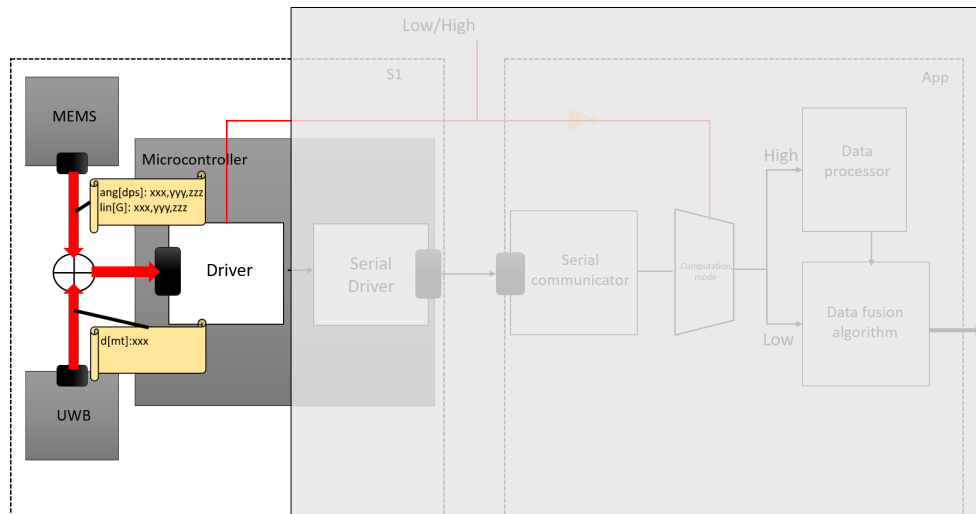


Figura 2.6: Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 1

I dati acquisiti sono "grezzi" e devono essere elaborati (vedi cap.4.1). Quando e da chi questa computazione verrà eseguita durante il flusso, viene stabilito attraverso il comando Low/High. Da questo ne consegue anche la modalità di funzionamento del driver in:

- **HCM:** High Computation mode, i dati vengono elaborati dal microcontrollore
- **LCM:** Low Computation mode, i dati verranno elaborati in seguito dal sottosistema *App*

La scelta tra quale di queste due modalità utilizzare verrà motivata nel capitolo riguardante l'analisi dei risultati (cap.4), per il momento ipotizziamo di settare la linea di comando al valore "Low" e quindi di utilizzare il driver in *LCM*. Con queste impostazioni i dati grezzi vengono impacchettati ed etichettati con un timestep relativo, prima di essere inviati dal modulo *Serial driver* e ricevuti dal sottosistema *App* mediante il modulo *Serial communicator*, quest'ultimo non farà altro che inoltrarli all'ingresso del multiplexer *Computation mode* come mostrato in figura:

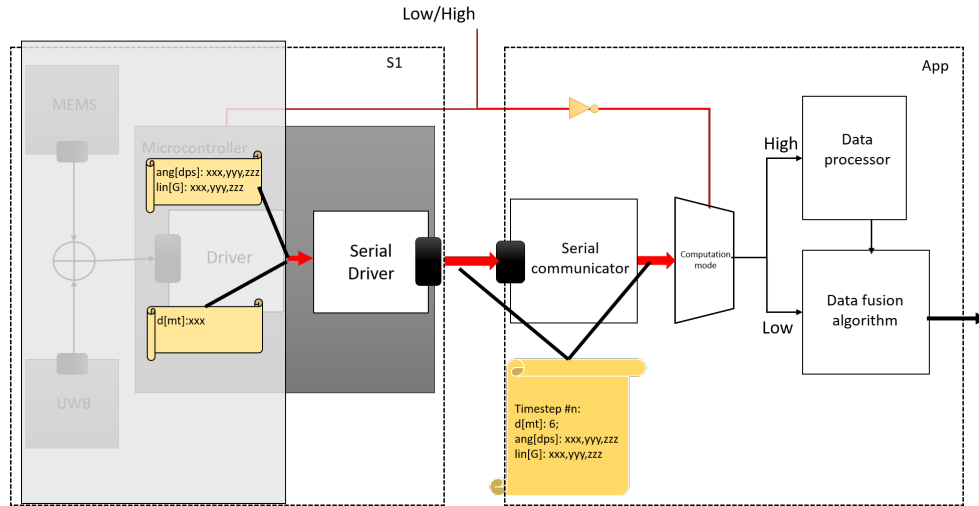


Figura 2.7: Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 2

Poiché per ipotesi abbiamo scelto di settare la linea di comando sul valore **Low**, la sua negazione sul ramo del sottosistema *App* fa sì che il multiplexer devii il flusso di dati verso il modulo *Data processor* come mostrato in figura:

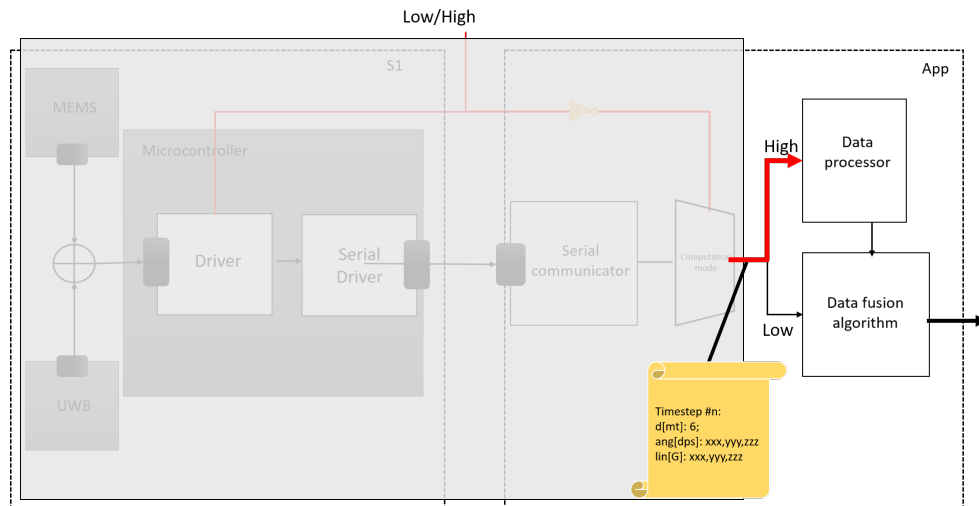


Figura 2.8: Rappresentazione del flusso di dati tra i moduli dei sottosistemi, step 3



## **Capitolo 3**

### **Tecnologie utilizzate**

#### **3.1 Sensori MEMS**

#### **3.2 Ultra-wideband**



## **Capitolo 4**

### **Elaborazione dei dati acquisiti e algoritmo di data fusion**

**4.1 Rumore dei dati grezzi**

**4.2 Tecniche di elaborazione**

**4.3 Algoritmo di data fusion**

## **Capitolo 5**

### **L'implementazione**

## **Capitolo 6**

### **Analisi e validazione dei risultati**

## **Capitolo 7**

### **Conclusioni e prospettive future**